Claims

1. A method for allocating space among a plurality of queues in a buffer, comprising:

sorting all the queues of the buffer according to size, thereby to establish a sorted order of the queues;

selecting at least one group of the queues consisting of a given number of the queues in accordance with said sorted order;

allocating a portion of the space in the buffer to said at least one group, responsive to the number of the queues in said at least one group; and

accepting a data packet into one of the queues in said at least one group responsive to whether the data packet will cause the space occupied in the buffer by the queues in the group to exceed the allocated portion of the space.

2. A method according to claim 1, wherein selecting said at least one group comprises selecting for inclusion in said at least one group the queues that are largest among said plurality of queues.

3. A method according to claim 2, wherein allocating the portion of the space comprises setting a size of the space proportional to a sum of a harmonic series.

4. A method according to claim 3, wherein setting the size of the space comprises establishing a total maximum buffer size B_k for the k largest output queues, wherein B_k is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^{k} \frac{1}{i}$$

wherein M is the total buffer space available to all queues, N is the number of output queues, and *i* is an integer index.

5. A method according to claim 4, wherein selecting the number k of the queues comprises selecting a fixed number of the queues.

6. A method according to claim 4, wherein k has a variable value.

7. A method according to claim 6, wherein k and B_n are set in accordance with the equation:

$$B_k = (n + \frac{m}{c^{n+1}}) \bullet \frac{M}{\log_c N}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that m and k satisfy the conditions

$$0 \le m \le c^{n+1}$$

and

$$k = \sum_{i=1}^{n} c^i + m .$$

8. A method according to claim 2, wherein buffer space is allocated within said at least one group of queues in accordance with a set of conditions which functionally define a known buffer management policy.

9. A method according to claim 8, wherein said known buffer management policy is chosen from a group consisting of complete partitioning (CP), sharing with maximum queue lengths (SMXQ), sharing with minimum allocation (SMA), sharing with maximum queue length and minimum allocation (SMQMA), and dynamic threshold (DT).

10. A method according to claim 2, wherein said given number is fixed.

11. A buffer management policy according to claim 2, wherein said given number and said portion of the space in the buffer are variable.

12. A method for allocating space among N output queues in a buffer of size M, which comprises:

sorting all the queues of the buffer according to size, thereby to establish a sorted order of the queues;

selecting a number k of said N output queues in accordance with said sorted order;

establishing a total maximum buffer space of B_k for said number k of said N output queues, wherein $B_k < M$;

ascertaining whether acceptance of an arriving packet destined for one of said number k of said N output queues will cause the space in the buffer used by said k queues together to exceed B_k; and

if the acceptance of an arriving packet destined for one of said k queues will cause the space used by said k queues together to exceed B_k, rejecting said packet.

13. A method according to claim 12, wherein said selecting a number k of said N output queues comprises selecting a number k of said N output queues that are largest among said N output queues.

14. A method according to claim 13, wherein establishing the total maximum buffer space comprises setting the maximum buffer size such that B_k is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^{k} \frac{1}{i}$$

wherein M is the total buffer space available to all queues, and N is the number of output queues, and *i* is an integer index which refers to an ordinal ranking of the queue size.

15. A method according to claim 13, wherein the buffer space is allocated within said number k of said N output queues in accordance with a set of conditions which functionally define a known buffer management policy

16 A method according to claim 15, wherein said known buffer management policy is chosen from a group consisting of complete partitioning (CP), sharing with

maximum queue lengths (SMXQ), sharing with minimum allocation (SMA), sharing with maximum queue length and minimum allocation (SMQMA), and dynamic threshold (DT).

17. A method according to claim 12, wherein selecting the number k of the queues comprises selecting a fixed number of the queues.

18. A method according to claim 12, wherein the value of k is variable.

19. A method according to claim 18, wherein the values of k and B_n are set in accordance with the equation

$$B_k = (n + \frac{m}{c^{n+1}}) \bullet \frac{M}{\log_c N}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that m and k satisfy the conditions:

$$0 \leq m \leq c^{n+1}$$

and

$$k = \sum_{i=1}^{n} c^i + m.$$

20. A shared memory switch comprising:

a memory providing buffer space of size M, which is adapted to contain a plurality of output queues; and

a controller, coupled to sort all the queues of the buffer according to size, thereby to establish a sorted order of the queues, and to allocate the space in the buffer to the output queues in accordance with said sorted order such that a portion of the space is allocated to a group of the queues consisting of a given number of the queues that are largest among the plurality of the queues responsive to the given number of the queues in the group.

21. A switch according to claim 20, wherein said portion of the space is allocated by setting a size of the space proportional to a sum of a harmonic series.

22. A switch according to claim 21, wherein said portion of the space is allocated for the k largest output queues and the total maximum buffer size B_k for said k largest output queues is substantially given by

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^{k} \frac{1}{i}$$

wherein *i* is an integer index which refers to an to ordinal ranking of the queue size.

23. A switch according to claim 22, wherein said portion of the space is allocated for the k largest output queues and the total maximum buffer size B_k for said k largest output queues is substantially given by the equation

$$B_k = (n + \frac{m}{c^{n+1}}) \bullet \frac{M}{\log_c N}$$

wherein n, c and m are parameters such that $1 < c < N$, and n is the largest integer such that m and k satisfy the conditions

$$0 \le m \le c^{n+1}$$

and

$$k = \sum_{i=1}^{n} c^i + m.$$

24. A shared memory switch according to claim 20, wherein said switch has N output lines and N corresponding output queues, wherein said given number is denoted by k and said portion of the space available for the k largest output queues is denoted by B_k, wherein $B_k < M$.

25. A shared memory switch according to claim 24, wherein B_k is substantially given by the equation

$$B_k = \frac{M}{(\ln N + 1)} \sum_{i=1}^{k} \frac{1}{i}$$

wherein i is an integer index which refers to the ordinal ranking of the queue in terms of the queue size.